INF

Studiengang
Medien- und
Kommunikationsinformatik

Hochschule Reutlingen
Reutlingen University

Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

Informatics Inside:

Human-Centered Computing

Informatik-Konferenz an der Hochschule Reutlingen
30. April 2014

ISBN 978-3-00-045427-1

Impressum

Anschrift:
Hochschule Reutlingen
Reutlingen University
Fakultät Informatik
Medien- und Kommunikationsinformatik
Alteburgstraße 150
D-72762 Reutlingen

Telefon: +49 7121 / 271-4002
Telefax: +49 7121 / 271-4042

E-Mail: infoinside@reutlingen-university.de
Internet: http://www.infoinside.reutlingen-university.de

Hochschule Reutlingen
Reutlingen University

Herstellung und Verlag: Hochschule Reutlingen
ISBN 978-3-00-045427-1

Inhaltsverzeichnis

Gestenerkennung & Augumented Virtuality

Thomas Bauer
Anforderungsanalyse zur computergestützten Erkennung der Deutschen Gebärdensprache *8*

Matthias Gutekunst
Augmented Virtuality zur Steigerung der Immersion in virtuellen Umgebungen *26*

Stanislas Mauser
Analysis of Finger- and Palm-based interaction paradigms for Touch-Free Gesture-Based Control of Medical Devices with the Leap Motion Controller *34*

Softwaretechnik

René Mangold
Selektion von Szenarien zur Optimierung von Simulationen im präventiven Krisenmanagement *46*

Arkadius Weister
Language Oriented Programming: Modulare domänenspezifische Sprachen *54*

Entwicklung Mobiler Anwendungen

Olaya De la Rosa Avitia
Strategy to Test Mobile Apps *70*

Viktoria Hoffmann
Optimierung der Usability von digitalen Fahrtenbüchern durch automatisches Erfassen von fahrzeugspezifischen Daten *80*

Johannes Kartheininger
Vergleich der Single Sign On Verfahren SAML und OpenID Connect *92*

Virtuelle Welten

André Antakli
Umgebungswahrnehmung von agentenbasierten simulierten Menschmodellen in virtuellen Welten im Kontext C3D *100*

Strategy to Test Mobile Apps

Olaya de la Rosa Avitia
Reutlingen University
Olaya.de_la_Rosa_Avitia@Student.
Reutlingen-University.DE

Abstract
Nowadays the development of a mobile app implies challenges and difficulties, which have to be faced by mobile app developers. Innovations lead to a rapidly evolving mobile app market, therefore apps should be developed faster and offered in short release cycles to the market. Testing is a decisive activity within the development process that helps to improve the quality of the app. This research paper describes a strategy to test mobile apps that overcomes the challenges that mobile apps confront and permits to test the app in a structural test environment.

Keywords
Mobile Software, Test Strategy, Software Testing, Mobile App, Mobile Devices, Mobile Computing, Cloud Computing.

CR-Categories
D.2.5 [**Software Engineering**]: Testing and Debugging—*Testing tools*. K.6.3 [**Software Management**]: Software process

---

Betreuer Hochschule: Prof. Dr.- Ing. habil. Natividad Martínez Madrid
Hochschule Reutlingen
Natividad.Martinez@Reutlingen-University.de

Betreuer Firma: Carlos Orozco
Robert Bosch GmbH
Carlos.Orozco@de.bosch.com

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz 2014
30. April 2014, Hochschule Reutlingen

1 Introduction
As predicted by analysts, the number of users that access the Internet on mobile devices has exceeded the users on desktops. Users are changing desktops for mobile devices such as laptops, smartphones, tablets, glasses and smart watches [1]. As a consequence of this tendency companies are modifying the business models and creating a mobile strategy to react properly to those changes. This strategy includes often the adaptation of websites to be shown adequately on mobile devices, development of mobile apps, launching of mobile campaigns and location-based services.

This mobile business strategy is different to the traditional. Nguyen [2] mentions that because of the unique characteristics of mobile systems, mobile applications are usually more complicated to develop than non-mobile applications. This complexity represents a challenge for mobile developers, especially to those who develop for Android, which is highly fragmented: Hundreds of mobile devices exist, the hardware and screen resolutions are different and the components in the devices change constantly. Therefore as Nguyen mentioned, testing if a mobile app is going to work adequately in each device is a complex task.

Because of the different characteristics of mobile devices, specific testing methodologies and concepts are also necessary to evaluate the functionality of a mobile app [11].

But for this evaluation there are other options to test apps without the necessity of having each device. As Ridene [3] mentioned, acquiring hundreds of mobile devices to implement repetitive and large-scale test activities in an industrial way causes problems in terms of effort, quality assurance and cost.

This paper provides answers to the following questions:

- Which are the attributes that characterize mobile computing and differentiate it from traditional computing?
- How do the architecture and usage context of an app influence the test cases?
- What are the fundamental challenges on testing mobile apps?
- Which tools and concepts can be used to efficiently test an app?
- When is it necessary and reasonable to automate the tests?

This research paper is structured as follows: Section 2 introduces the mobile app testing topic taking in consideration the app background based on the context of use and the architecture. In section 3 the challenges that mobile app developers confront are described. The test execution process, with the testing types and automation testing are addressed in section 4. The strategy to test mobile apps is presented in section 5 and finally the conclusion of the research paper is shown in section 6.

2 Mobile apps testing

Haller describes in [1] that the process to develop mobile apps is not suitable to the classical waterfall development approach: Start of the project with the definition of specifications and requirements that the software has to satisfy, after that the implementation of these specifications by the developers. Then testing what was implemented in comparison with the specifications and finally if the tests are successful, the customer pays.

During the testing phase, the app has to be tested by developers and testers to check that it works correctly and fulfills the requirements. But according to Haller, from a business point of view the success of the app depends also on the answers to:

- Is it what the users expect?
- Are the offerings of the competition good?
- Does the app boost the reputation of a company? (Figure 1)

Figure 1: App quality and success in the mobile world [1]

These three questions help to recognize how the quality of an app is perceived by the users, therefore testing has to be more than the definition of test cases and their corresponding repetitive implementation [1]. Haller [1] presents a series of points that a tester has to know before the test cases are determined:

- **Context of use:** In which group is the app user? (repetitive now, bored now or urgent now)
- **Architecture:** Is it a native client/server app, native standalone app, mobile web app or a launcher app?
- **User base:** Is the app for an open market or only for a company?

After the information is available, the test type will be applied and the development stage in which the solutions will be used, can be defined. Other factors and influences are: Timeline of the project, complexity, cost/benefit analysis and the existing infrastructure. The following subsections

describe the first two aspects that Haller considers are important to define the test cases.

2.1 Context of use

The context of use is the environment and actual conditions in which an app is used and it provides a basis for the later design of the test cases. First, it is necessary to understand which mobile user group will use the app based in Leland Rechis' classification [4]:

- **Repetitive now:** Is the group of users that use a mobile device to do the same over and over. This could be checking for new emails, looking for the weather forecast or reading the news.
- **Urgent now:** Are the users who need information immediately, for example a train timetable.
- **Bored now:** These users are for example waiting for someone or sitting in the train and have time to spend.

Haller [1] states that once the usage group was identified, the context of use of an app can be defined answering three questions:

- What is the app type? The app can be interactive (for example a game) where the visuals and graphics are important or a no-interactive app like online shops or phone directories with a straight-forward design.
- How long is the app going to be used? It can be hours (playing a game), minutes (consulting a map) or just seconds (phone book).
- How frequent does the mobile device interact with back-end servers? An app can require data download and therefore needs a stable server connection.

For example an app that shows to the user when a hop-on hop-off tourist bus will arrive is within the user group urgent now. The answers to the three questions presented before are: The app is non-interactive, it is used only for a couple of minutes and does not have to interact with back-end servers since the information is downloaded together with the app and stays static (delays in the tourist bus are not shown). This is the description of the environment or scenario in which the app is used.

2.2 Architecture

If the developers and testers understand the architecture of the app, it will be easy to identify which part can fail and hence where it is necessary to implement tests. In the mobile world, there are four main architectural patterns [1]:

- **Native standalone apps:** These are apps that are preinstalled on a mobile device and don't have to communicate with a server, for example the calculator or the alarm.
- **Mobile web apps:** These are web pages optimized to be shown and run code in a web browser on mobile devices.
- **Native client server (C/S) apps:** Are native apps that were downloaded and installed by users. The app retrieves data from a back-end server, for example a mobile banking app.
- **Launcher app:** Stand between native C/S apps and mobile web apps: The app is downloaded and installed on the device by the users and provides some features and/or can redirect to a mobile web app.

3 Challenges in testing mobile apps

In section one, the Android fragmentation was mentioned. This represents a problem for the developers and is at the same time an advantage for users [5] as users like to have a large pool of devices they chan choose from. Only in 2013 about 12000 device types were registered in OpenSignal[1], a company specialized in mapping the mobile internet.

Despite of the slogan "write once, run everywhere" (W.O.R.E.) created by Sun Microsystems to define the benefits of Java,

[1]http://www.macerkopf.de/2013/07/31/android-fragmentation/

the practice shows that adjustments are necessary to run an app on a different device that responds in the same way and has a consistent behavior [3].

Mobility is characterized and differenced by the following constraints: Mobile elements are resource-poor, mobile devices are more vulnerable, and performance of connectivity is highly variable with a limited energy source. Since the mobile information systems are different to the traditional market, users need apps with an intuitive navigation, easy to use, with concise text, efficient and suitable for activities on the road [6].

The challenges that developers have to deal with taking in consideration the characteristics and differences of mobile apps arise according to Baride and Dutta [8] from four main areas:

1.- Diversity of the device environment: Different devices make the reusability and maintenance of test cases more difficult [9]. One system targets multiple devices and all the devices have to be considered for the tests, which might be difficult to carry out and expensive. Tablets using the same app with the same environment may have different responses caused by sensors that are calibrated differently for each device.

2.- Hardware configuration and network-related changes: The devices have to interact with a dynamic environment: Connectivity, bandwidth and networking considerations [10] as well as wireless signals [9]. Compo-nents are upgraded and the configuration changes constantly, how does a testing environment which can tackle these challenges have to look like?

3.- Rapid application development (RAD) methodologies: The short innovation cycles for hardware and platforms generate the necessity to reduce the time taken for the development and testing cycle without compromising coverage and the quality of the app [1].

4.- Device limitations: Some restrictions are mentioned below, although many device limitations could affect more a smartphone than a tablet:

- Processing ability
- Memory capacity
- Communication ability [9]

4 Testing methodology

For the development of high quality software products a well-defined software development process is necessary [5]. Testing helps to identify if the requirements defined at the beginning are fulfilled in the end product and it reduces the possibility that the software reacts undesirably. It saves costs and failure in the field.

Baride defines in [8] the factors that determine if a mobile testing program is successful. These are: Complexity of the app, testing for a mobile environment and the use of test automation, emulator and actual devices.

Mobile apps have to go through the following testing methodology: First the definition of the test strategy where the test plan with its approach, risks, contingencies, recommendations and resource requirements (schedule, tools, roles and responsibilities) are described. Then the specification of the test design contains features that have to be tested, scenarios in which the app is used and the acceptance and release criteria. After that the test cases are defined with the test actions, input data, execution conditions and expected results. Finally the summary report provides the information covered by the tests and the accomplished results after the test execution.

The following sections describe the different ways, in which a test can be executed, what the types of testing are and when it makes sense to implement the testing automation.

4.1 Test execution

There are different ways and solutions to carry out tests of mobile apps. These will be described in the following subsections.

4.1.1 Crowd-sourced mobile app testing

It consists of a community of common mobile users that get money for testing an app. The disadvantage here is that the testers are not professional, which represents difficulties and misunderstandings when the non-technical feedbacks have to be interpreted [3].

4.1.2 Emulators

An emulator simulates the functions of a mobile device on a laptop or PC, allowing to check the app during design and development[2]. Emulators are designed for a specific platform and can even simulate a camera or incoming text messages [1]. Nevertheless emulators don't emulate all devices, network availability and speed, cache and stack size, memory limitations and the speed of content rendering on a specific device [8].

4.1.3 Local device

It consists of testing the app manually on the target mobile device [12] to check that at runtime in the user's hand the software has the expected behavior [3]. Usually the testers own the device and it might or might not be connected to the PC. The tester can also use tool support like eye tracking, recording software or automated tests e.g. Experitest[3] [1]. Using adequate testing techniques maximizes the coverage of device diversity without the need of acquiring each device [13].

4.1.4 Private device cloud

It is a company-internal, centralized pool with devices connected to web servers and network operators to analyze how an app behaves with events like SMS, calls and low battery. The user experience and audio outputs can be captured [3]. The device can be selected and used as a local device; the keys are electrically wired so that a tester can realize inputs (screen touch, unplug charger or USB) using a PC [14] [3]. The benefits of having a private device cloud are that a team takes care of device issues and has control over the infrastructure. Also if a device doesn't work, testers can switch to another one.

4.1.5 Public device clouds

The difference with the private device clouds is that for public device clouds the access is offered by service providers through the Internet. One of the disadvantages is the integration of the pool devices into an already existing test infrastructure [1]. For example Samsung offers a Remote Test Lab (RTL[4]), which allows testing apps on different Samsung devices for a certain number of hours per day for free.

4.2 Types of tests

Jayamalrao [15] presents in figure 2 a cycle to test mobile apps:

Figure 2: Mobile testing types [15]

1.- Sanity test: Proves the correct functionality of the app. It is the base for further tests.

2.- Functional/UI: Evaluates if the app is working as expected with the documented

[2] Using the android emulator: http://developer.android.com/tools/devices/emulator.html

[3] http://experitest.com/

[4] Samsung RTL http://developer.samsung.com/remotetestlab/rtlDeviceList.action

requirements. Test the text, logos, and usability aspects on the UI level.

3.- Interruption testing: Checks if the app reacts as expected during interruptions like incoming calls, low battery, zero network, etc.

4.- Regression testing: To ensure that changes and new features don't affect the basic functionality of the app.

5.- Stress/Performance: Analyses which is the response time and performance while navigating between different screens. Determines if the app is secure in terms of session handling, data transmission, storage, etc.

6.- Localization testing: Implementation of the same tests in a local language with translation of the interfaces, documentation and support.

7.- Field testing: Are explorative tests which have the purpose to test an app in the usage context and real world in which users will use the app. For example a hiking app must be carried to the top of mountains and valleys to test the correct functionality [1].

4.3 Test automation

There are different reasons why mobile test cases should be automated: Many test cases have to be executed many times and frequently using different devices to detect bugs early. In such cases, it is not possible for testers to keep focus and execute the test cases for a long period. Therefore automation plays a primordial role here [1]. Repeated testing can reduce testing effort, costs and workload of developers and testers [6], these also accelerate and facilitate the development and testing process. Test scripts cover the basic functionalities of the app and run on a predetermined time and collect useful data. With automated compatibility tests, testers can be sure that an app runs on relevant devices [1] and test scenarios can be adapted or reused for similar apps [3].

The use of automated tests makes more sense for non-interactive apps than for interactive apps since it is more difficult to put the physical interaction in a script, for example in sport simulation apps [1].

An open-source tool for Android apps testing is Robotium[5] that permits to write and execute automated tests of activities, menus, dialogs and context menus for Android [16]. In Lesspainful[6] it is also possible to implement automated app testing on its device cloud.

5 Strategy to test mobile apps

In the previous sections, the areas involved in mobile app testing were described. What has to be tested can be defined by a test strategy for mobile testing.

Figure 3 shows the strategy to follow when testing mobile applications. As an example the app Bosch Events[7] will be used. This app is used to view information of an event at Bosch (schedule, location, speaker infor-mation, exhibitors), to take photos of the event and upload them, see who is going to attend the event, send feedback to the speakers and see likes, comments and trends of the attending people. This is a native mobile app which is used by the user group repetitive now. The platform where the app is used is iOS (5.1 and later) and Android (version 2.3.3 and up).

For the second phase (test strategy) the suitable test tool to use in the first phases of testing is an emulator since it facilitates and speeds up the early resolution of problems. Emulators also cover GUI aspects like the size of the screen. For testing physical interactions and network effects, local devices will be used. The selected devices represent those who are used more like the iPhone and Samsung Galaxy. In this case only the device and emulator approach will be used.

In the test design phase the test scenarios and test cases are defined. For this a table can be

[5] http://code.google.com/p/robotium/

[6] https://www.lesspainful.com/

[7] https://itunes.apple.com/app/id726491006

used where all the information can be systematically tested (Table 1):

Table 1: Sample test scenario

Require-ment	Test scenario	Test cases
Login	OK Cancel	1. Verify ok with valid data 2. Verify ok with invalid data 3. Verify cancel
Search	Search Cancel	1. Verify Search with valid data 2. Verify Search with invalid data 3. Verify Cancel
Add comment	OK Cancel	1. Verify OK after entering a string 2. Verify OK with invalid data 3. Verify Cancel

In the test execution, the different test cases are tested first with an emulator and then with local devices. If potential defects are detected, they are analyzed and prioritized. Then changes are made to fix the defects which are tested again to check that these were fixed. Finally a detailed report with the results of the test process is written.

6 Conclusion

The objective of this research paper was to describe a strategy to test mobile apps and to provide a basis knowledge that helps with the implementation of this test. It also offered answers to the questions in section 1.

This paper described first in which context scenarios an app can be used and the different architectures for apps. After this, the inconveniences imposed by the different challenges related to testing mobile apps and the peculiarities about mobile testing were discussed. Diverse device specific challenges were: The device fragmentation syndrome, rapid application development (RAD) methodologies, hardware configuration and network-related changes.

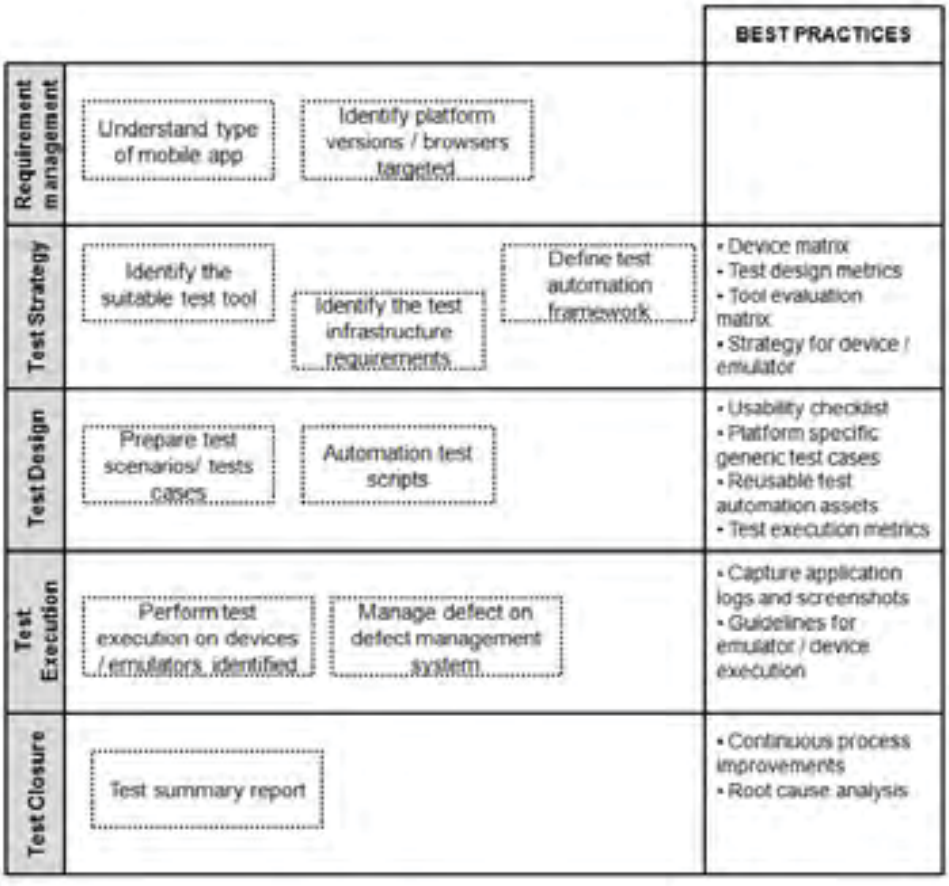

Figure 3: Mobile Testing – Process and Methodology [15]

The various solutions to test mobile apps were described and the types of tests as well. The topic test automation was approached and the advantages of automating test cases were mentioned.

Finally a strategy to test mobile apps was presented. This strategy aims to set a reference structure within the development process that identifies which solutions can be applied in each phase of the software process.

The testing process doesn't end with the release and publish of an app on the market. The app has to be continuously tested on new devices, new hardware conditions and OS versions. The market changes steadily and offers actually a great variety of testing tools like Robotium or Lesspainful. However testing still has many limitations, for example some tests cases depend on human sense, like color matching. These tests can't be implemented in automated tests and the results aren't easily quantized [6].

Therefore, mobile apps testing still needs improvements and solutions that enable cost-effective testing. But as Haller [1] writes: "*An app's market triumph requires more: Understanding the user, design and usability, and creativity and innovation*", not only in defining test cases.

7 References

[1] K. Haller. Mobile Testing. Swisscom IT Services. ACM SIGSOFT Software Engineering Notes. November 2013 Volume 38. Online available in: http://doi.acm.org/10.1145/2532780.2532813. Accessed on: March 7th, 2014

[2] M. D. Nguyen, H. Waeselynck and N. Riviere. Testing Mobile Computing Applications: Toward a Scenario Language and Tools. Workshop on Dynamic Analysis, July 21th, 2008. ACM 978-1-60558-054-8/08/07

[3] Y. Ridene and F. Barbier. A Model-Driven Approach for Automating Mobile Applications Testing.ECSA '11 Workshop SAVA '11. September 13, 2011. Essen, Germany. ISBN 978-1-4503-0618-8/11/09

[4] S. Wellman. Google lays out its mobile user experience strategy. Information Week April 11 2007. Online available in: http://www.uxmatters.com/mt/archives/2011/05/three-layers-of-mobile-user-experience.php#sthash.2l4qTSg3.dpuf accessed on: March 4th, 2014

[5] M. E. Delamaro, A. M. R. Vincenzi and J. C. Maldonado. A Strategy to Perform Coverage Testing of Mobile Applications. International Workshop on Automation of Software Test (AST '06) May 23th, 2006 Shanghai, China.

[6] C. Wuthrich, G. Kalbfleisch, T. Griffin, and N. Passos. On-line Instructional Testing in a Mobile Environment. JCSC 18, 4. Midwestern State University. April 2003.

[7] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, ser. PODC '96. New York, NY, USA: ACM, 1996, pp. 1–7. Online available in: http://doi.acm.org/10.1145/248052.248053, accessed on: March 3th, 2014

[8] S. Baride and K. Dutta. A Cloud Based Software Testing Paradigm for Mobile Applications. ACM SIGSOFT Software Engineering Notes. May 2011 Vol. 36 Nr. 3

[9] J. Bo, L. Xiang, G. Xiaopeng: MobileTest: A Tool Supporting Automatic Black Box Test for Software on Smart Mobile Devices. 2nd International Workshop on Automation of Software Test (AST'07), Minneapolis, MN, 20–26 May 2007

[10] Y. Dubinsky and A. Abadi. Challenges and Research Question for Testing in Mobile Development. MobileDeli '13, October 28, 2013 Indianapolis, Indiana USA.

[11] Q. H. Mahmoud. Testing wireless java applications. Sun Microsystems, on-line article, Nov. 2002. Online available at: http://developers.sun.com/techtopics/mobility/midp/articles/test/, accessed on: December 10th, 2005

[12] I. Satoh. A testing framework for mobile computing software. IEEE Transactions on Software Engineering, 29(12):1112–1121, December 2003.

[13] H. Muccini, A. Di Francesco, P. Esposito: *Software Testing of Mobile Applications: Challenges and Future Research Directions*, 7th International Workshop on Automation of Software Test (AST '12), June 2–3, 2012, Zurich, Switzerland

[14] K. Heller. Mobile Testing. Swisscom IT Services. ACM SIGSOFT Software Engineering Notes November 2013 Volume 38. Online available in: http://dl.acm.org/citation.cfm?doid=2532813, accessed on: March 3th, 2014

[15] G. Jayamalrao. Cognizant 2011. Online available in: http://www.qaac.org/wp-content/uploads/2012/07/Mobile-Testing-QAAC.pdf, accessed on: March 5 th, 2014.

[16] S. Diewald, L. Roalter, A. Möller and Matthias Kranz. Technische Universität - München. MUM '11, Dec 7-9 2011, Beijing China. ACM 978-1-4503-1096-3/11/12.

[17] L. Zhifang, L. Bin, G. Xiaopeng: Test Automation on Mobile Device, AST'10, May 3-4 2010, Cape Town, South Africa

[18] S. Waterson, J. A. Landay and Tara Matthews. In the lab and out in the wild: remote web usability testing for mobile devices. Conference on Human Factors in Computing Systems, 2002, pp. 296–297.